An Implementation of Segmented Discourse Representation Theory *

Nathalie Daver
daver@irit.fr

Abstract

SDRT is a sophisticated logical discourse theory for discourse analysis that integrates two traditions: dynamic semantics and discourse structure theory because it turns out that it is necessary to model the interaction between the semantic content of texts and their structure to give an adequate account of several discourse phenomena. The implementation of SDRT is needed to test the validity of the axioms and rules expressed in that interesting framework. In that paper, we will focus on the implementation of the logical inference engine.

1 Introduction

Dynamic Semantics (e.g. Kamp's Discourse Representation Theory (1993), Groenendijk and Stokhof's Dynamic Predicate Logic (1991)), in which the meaning of a sentence is a function from discourse contexts to discourse contexts, appeared in the eighties because the Montagovian framework was not well suited to the analysis of intersentential temporal and pronominal anaphora. But these formal theories failed to take into account the discourse relations (*narration*, *explanation*, ...) that hold between discourse segments, and the hierarchical structure that these discourse relations impose on the discourse.

Another tradition of studies on discourse has concentrated on the description of the hierarchical structure of discourse. But, in general, these theories (Mann and Thompson's Rhetorical Structure Theory (1987), Hobbs's TACITUS (1991), Scha and Polanyi's Linguistic Discourse Model (1988), Grosz and Sidner's theory (1986)) did not link their accounts of discourse interpretation to a detailed analysis of intra-sentential semantics.

Because of the interaction between semantics and discourse structure, none of the approaches mentioned above can give an adequate account of discourse structure or discourse content on its own. By combining these two

*I am grateful to Nicholas Asher, Myriam Bras, Mario Borillo, Laure Vieu for their helpful comments and advice. Thanks to the two anonymous reviewers for their comments.

research areas, Asher's (1993b) Segmented DRT provides a better suited framework to the analysis of:

- pronominal anaphora since discourse relations impose a hierarchical discourse structure which plays an important role in determining what antecedents of a pronoun are available

- truth conditional content of discourse because it is affected by discourse relations

- some semantic problems, like lexical disambiguation, determination of the temporal and spatio-temporal structure of discourse as it turned out that the semantic information that can be derived from the syntax is not sufficient to give an adequate account of those problems. The solutions given by Dynamic Semantics can be improved upon by considering those discourse features as semantic effects of discourse relations which are computed from various information sources.

SDRT is expressed in a nonmonotonic logic, Commonsense Entailment (Asher and Morreau, 1991), which has all the specific properties required by discourse analysis. The implementation of SDRT consists therefore in implementing the logic but also the axioms and rules expressed in that logical framework. In this paper, we will focus on the implementation of the logic, which is the main problem involved. Other aspects of the implementation can be found in (Daver, 1994).

2 SDRT

2.1 A brief review

SDRT extends Kamp's DRT (1993) to represent the discourse relations (*narration*, *explanation*, ...) that hold between discourse segments. The representation of a text is not a DRS but rather a complex structure called a segmented DRS, in which DRSs are related by discourse relations. Some of the discourse relations are subordinating (e.g. *elaboration*), which is why the structure has the form of a tree. Here is a simple example from Lascarides and Asher (1991):

Example 1

1. John had a fantastic meal.

2. He ate salmon.

3. Then he won a dancing competition.

$\mathrm{DRS}_a - \mathrm{DRS}_c$ $\vert$ DRS_b
$elaboration(a, b)$ $narration(a, c)$

The SDRS is built up recursively by adding each new sentence to the SDRS already built up from the analysis of previous sentences. The integration of a sentence can be decomposed into four steps: first of all, we compute the DRS which corresponds to that sentence. If the text is coherent, that DRS must be related by at least one discourse relation, to at least one constituent, according to constraints that define what constitutes an acceptable attachment. We thus determine, in the second step, the set of constituents which meet the attachment constraints. Then, we must compute the discourse relations that hold between the current sentence and one of the acceptable constituents. The attachment of a new constituent by means of a particular discourse relation may have effects upon the truth-conditional content of the constituents. Thus, after attachment, constituents have to be revised.

SDRT is expressed in a nonmonotonic logic, Commonsense Entailment (CE), and in particular discourse relations are computed in that logic. Given that discourse relations computation is a very important part of SDRT, it will serve to illustrate calculations in CE.

2.2 Inferring discourse relations nonmonotonically

The presence of clue words sometimes suffices to compute the appropriate discourse relation, but not always. Often, we must also exploit information about the semantic content of the constituents, pragmatic principles and domain knowledge. But, even with all these information sources, we are still for the most part making defeasible inferences as to what discourse relation the author intended. Thus the underlying logic for this computation must be a nonmonotonic logic.

Lascarides and Asher (1991) show that if we want to characterize discourse relations in terms of defeasible rules, we have to use a defeasible reasoning system which (i) supports the patterns of inference listed below, (ii) solves the problem of irrelevance and (iii) can reason with embedded defaults, i.e. default rules where one default rule is embedded in another (e.g. $(a > (b > c))$). This is why they choose CE, presented in (Asher and Morreau, 1991), since Asher and Morreau (1991) proved that CE meets all these requirements in a satisfactory way.

Defeasible Modus Ponens If birds fly and Tweety is a bird, then Tweety flies.

Nixon Diamond if Nixon is a Quaker and Nixon is a Republican and Republicans are non-pacifists and Quakers are pacifists then no conclusion is inferred (neither Nixon is pacifist nor Nixon is non-pacifist).

Penguin Principle If Tweety is a penguin and birds normally fly and penguins normally don't fly and penguins are birds then Tweety doesn't fly.

The language of CE is that of first order logic, augmented with a nonmonotonic conditional operator $>$. $(A > B)$ means: if A then *normally* B.

The rules introduced below are examples of default rules, used to infer discourse relations for narrative texts. They also tell us which temporal relations the discourse relations entail. Let $\langle\tau, \alpha, \beta\rangle$ be the update function, which means "the representation τ of the text so far, of which α is an acceptable constituent, is to be updated with the representation β of the current clause via a discourse relation with α".

$$\langle\tau, \alpha, \beta\rangle > narration(\alpha, \beta)$$
$$\langle\tau, \alpha, \beta\rangle \wedge D\_permissible\_subtype(\beta, \alpha) > elaboration(\alpha, \beta)$$
$$\langle\tau, \alpha, \beta\rangle \wedge cause(\beta, \alpha) > explanation(\alpha, \beta)$$

The entailed temporal effects are:

$$narration(\alpha, \beta) \rightarrow e_\alpha < e_\beta$$
$$elaboration(\alpha, \beta) \rightarrow e_\beta \subseteq e_\alpha$$
$$explanation(\alpha, \beta) \rightarrow \neg(e_\alpha < e_\beta)$$

cause and *D_permissible_subtype* above are inferred from various knowledge sources in the logic.

The rules above mean that if (i) no information is derivable from world knowledge and lexical knowledge (i.e. if the only axiom applicable is the one of *narration*), and if (ii) the constituents α and β are to be attached in the SDRS τ, then *narration* is inferred by default. As indicated by the temporal effects of *narration*, the event described in α occurs before the one in β, so we assume that, by default, the text is orderly, i.e. the events are described in the right order. But, if the axiom for *explanation* is also applicable, then the *Penguin Principle* forces us to infer only *explanation*, since *narration* and *explanation* are not compatible, as indicated by the temporal effects. And, if the three axioms are applicable, then no discourse relation is inferred, because of *Nixon Diamond*. This is the case when the discourse is not coherent.

3 Commonsense Entailment

This logic defines two notions of consequence: $\models$, the monotonic one, and $\approx$, the nonmonotonic one.

3.1 The monotonic consequence

The semantics of the conditional operator is defined in terms of a selection function $*$ from worlds and propositions to propositions. As described in (Asher and Morreau, 1991), intuitively, $*(w, p)$ is the set of worlds where the proposition p holds together with everything else which, in world w, is *normally* the case when p holds. So, $*$ encodes assumptions about normality. The truth conditions of defeasible rules are defined as follows:

$$M, w \models \phi > \psi \text{ if and only if } *(w, [\phi]) \subseteq [\psi] \tag{1}$$

In words, the above says that if ϕ, then normally ψ is true with respect to the model M at the possible world w, if the set of worlds that defines what is normally the case when ϕ is true in w, contains the information that ψ is true.

To get the right nonmonotonic patterns of inference (Defeasible Modus Ponens, Penguin Principle and Nixon Diamond), the selection function has to be constrained in the following way[1]:

C1: *facticity*: $*(w, b) \subseteq b$
Facticity captures the intuition that one of the properties of a normal bird is that he is a bird.

C2: *specificity*: if $*(w, b) \subseteq f$ and $*(w, p) \cap *(w, b) = \emptyset$ and $*(w, p) \neq \emptyset$ then $*(w, b) \cap p = \emptyset$
Specificity encodes the constraint that normal birds are not penguins because penguins don't fly.

As far as the derivability notion is concerned, the axioms corresponding to *facticity* and *specificity* are the following:

A1: *facticity*: $(b > b)$

A2: *specificity*: $((p \to b) \land (b > f) \land (p > \neg f)) \to (b > \neg p)$

EXAMPLE 2
If

$$\begin{array}{lcl} M, w & \models & penguin(x) \to bird(x) \\ M, w & \models & bird(x) > fly(x) \\ M, w & \models & penguin(x) > \neg fly(x) \end{array}$$

[1] In (Lascarides and Asher, 1991), *specificity* has been replaced with *dudley*: $*(w, p \cup q) \subseteq *(w, p) \cup *(w, q)$ Accordingly, in addition to Defeasible Modus Ponens, Penguin Principle and Nixon Diamond, their logic also supports the following pattern of inference:

Dudley Doorite If Quakers are pacifists and Republicans are pacifists then Quakers or Republicans are pacifists.

The problems raised by the implementation of such a logic have led us to build a theorem prover for the version of CE presented in (Asher and Morreau, 1991) where the selection function is constrained by *facticity* and *specificity*.

then $*(w, bird(x))$ verifies the formulas:

$fly(x)$ because of the truth conditions of $(bird(x) > fly(x)))$

$bird(x)$ because of *facticity* and the truth conditions of
$(bird(x) \;>\; bird(x)))$

$\neg penguin(x)$ (because of *specificity* and the truth conditions of
$(bird(x) \;>\; \neg\; penguin(x)))$

and also all the formulas that hold when the formulas $fly(x)$, $bird(x)$, $\neg penguin(x)$ are verified (e.g. $fly(x) \wedge bird(x)$, $a \rightarrow fly(x) \ldots$).

3.2 The nonmonotonic consequence

A dynamic semantics, built on top of the truth conditional semantics, defines a nonmonotonic consequence relation, $\approx\!\!\!\!|$, which encodes plausible but defeasible inferences, that is, $\models\!\!\!\!\approx$ defines inferences that are plausible given that one knows only what is in the premises plus the laws of logic. The definition of $\models\!\!\!\!\approx$ exploits the canonical model M for $\mathcal{L}_{>}$, its set of worlds W_{can} and also four concepts: information states, the informationally minimal state, updating, normalisation. As defined in (Asher and Morreau, 1991), the informationaly minimal state is a set of worlds which supports only the laws of logic. Information states are sets of possible worlds obtained by updating the informationally minimal state with a set of premises: thereby, we identify the set of worlds that characterizes believing only the premises and the laws of logic. The normalisation function encodes in the semantics the notion of assuming everyone and everything as normal as possible.

Definition 1 (Normalisation)
Let δ be an element of the domain $\mathcal{D}_M$.
$[\phi]$ denotes the set of worlds in which δ has the property of being a ϕ.
$*(S, [\phi]) := \bigcup_{w \in S} *(w, [\phi])$
$[\phi] \setminus *(S, [\phi])$ is the set of worlds, where according to all of the worlds in S, δ is thought a ϕ, not a normal ϕ.

$$Normalisation(S, \phi) := \begin{cases} \{w \in S | w \notin ([\phi] \setminus *(S, [\phi]))\} & \textit{if } S \cap *(S, [\phi]) \neq \emptyset \\ S & \textit{otherwise} \end{cases}$$

In words, the normalisation function isolates those worlds in the set of informational states, S, where δ is a normal ϕ, if it is possible to assume δ to be a normal ϕ. This is the case in which $S \cap *(S, [\phi]) \neq \emptyset$.

Example 3
If S contains the premises: $bird(tweety)$ and $bird(x) > fly(x)$, then

1. $*(S, [bird(tweety)])$ verifies the formulas $bird(tweety)$ and $fly(tweety)$.

2. $S \cap *(S, [bird(tweety)]) \neq \emptyset$

Assuming Tweety to be a normal bird is possible in this case, so the normalisation function will isolate all worlds from S where Tweety is a normal bird and return those worlds where $bird(tweety)$ and $fly(tweety)$ hold.

If S contains the premises: $bird(tweety), \neg fly(tweety)$ and $bird(x) > fly(x)$, then

1. $*(S, [bird(tweety)])$ verifies the formulas $bird(tweety)$ and $fly(tweety)$.

2. $S \cap *(S, [bird(tweety)]) = \emptyset$

Assuming Tweety to be a normal bird is in this case hopeless, so the normalisation function simply returns the original state S.

In normalising with respect to many individuals and many properties, the normalisation function has to be iterated. The ordering in which the normalisations are performed affects the result, which is why all different orderings of the iterations are taken into account. For a theory Γ which does not contain any embedded default rules, the nonmonotonic consequence can be defined in the following way:

Definition 2
Let $ANT(\Gamma) = \{\psi : (\psi > \phi) \in \Gamma\}$. *Suppose that* $\mid ANT(\Gamma) \mid = n$.
The Γ*-Normalisation sequence for a given ordering* ζ *on* $ANT(\Gamma)$) *is defined as:*

$$\begin{array}{lcl} \Gamma_0(\zeta) & = & S \cap \Gamma \\ \Gamma_{i+1}(\zeta) & = & Normalisation(\Gamma_i(\zeta), p), \ \textit{for } \zeta(p) = i+1 \end{array}$$

Definition 3 (Non-monotonic consequence)

$\Gamma \approx\!\!\!\models \phi$ *iff* $\forall \zeta, \Gamma_n(\zeta) \models \phi$

4 Implementation of the logical inference engine

The semantic notion of nonmonotonic consequence is hard to work with, which is why we have chosen to exploit the proof theoretic equivalent for $\approx\!\!\!\models$, defined in (Asher, 1993a) by using the notion of extension. The first version of the implementation of CE, presented below, does not allow the use of embedded default rules. For the sake of concreteness, we will look at three examples.

EXAMPLE 4

D	C
$penguin(x) \rightarrow bird(x)$ $bird(tweety)$	$bird(x) > fly(x)$ $bird(x) > legs(x)$ $penguin(x) > \neg fly(x)$

Suppose that the theory Γ contains the set D of default rules and the set C of classical rules, listed above. Recall that, according to the normalisation process, Tweety is assumed to be a normal bird if possible. Therefore, in the extension calculus, $bird(tweety) \rightarrow bird(tweety) \wedge fly(tweety) \wedge \neg penguin(tweety)$ is added to C, that is, Tweety is assumed to be a normal bird if everything which is normally the case when Tweety is a normal bird (i.e. $bird(tweety)$, $fly(tweety)$, $\neg penguin(tweety)$), is consistent with C (i.e. if it is possible to assume Tweety to be a normal bird). The issue that arises in the consistency test involves determining what has to be tested.

Let NB be the set of all the formulas that hold in the worlds where Tweety is a normal bird. And let ψ be the conjunction of all those formulas. We will obviously not test whether ψ is consistent with C, because, on the one hand, NB is infinite, and, on the other hand, we don't need to test the consistency of each formula of NB with C to be sure that they are all consistent with C. It suffices to find a subset NB' of NB which is such that the conjunction ϕ of the formulas of NB' implies each formula of NB. Thereby, if ϕ, called the *prime implicate*, is consistent with C, then we are sure that ψ is consistent with C. More formally:

$$NB = *(w, [bird(x)])\psi = \bigwedge f, \forall f \in NB$$

$NB' \subseteq NB$ is such that

$$\phi = \bigwedge f, \forall f \in NB'$$
$$\forall f \in NB, \vdash \phi \rightarrow f$$

fact if ϕ is consistent with C then ψ is consistent with C

The nonmonotonic theorem prover is thus divided into two parts. In the first part, we compute the prime implicates, and, in the second part, we perform the consistency tests.

4.1 Prime implicates computation

In the first part of the theorem prover, from our set of default rules D, we build up an other set of default rules D' as follows: we first compute the set $ANT(D)$ of the antecedents of the default rules, in such a way that, if two rules have the same antecedent, then this antecedent occurs only once in the set $ANT(D)$.

Definition 4

$$ANT(D) = \{\psi : (\psi > \phi) \in D\}$$

$$\forall a, b \in ANT(D) : a \neq b$$

In Example 4, $ANT(D) = \{bird(x), penguin(x)\}$

After having computed the set $ANT(D)$, we then compute for each element a of $ANT(D)$, the corresponding prime implicate ϕ_a. Then, $a > \phi_a$ is added to D':

Algorithm 1

for all $a \in ANT(D)$ **do**
  compute ϕ_a
  add $(a > \phi_a)$ to D'
end for

So we get:

D'

$bird(x) > \phi_b$
$penguin(x) > \phi_p$

Actually, the default rules are encoded in such a way that we cannot find two rules with the same antecedent. The fact that birds normally fly, and that birds normally have legs, is encoded by $bird(x) > (fly(x) \wedge legs(x))$, but not by $bird(x) > fly(x)$ and $bird(x) > legs(x)$. Accordingly, it's not worth computing the set $ANT(D)$. Thus, Algorithm 1 is replaced by Algorithm 2:

Algorithm 2

for all $(a > b) \in D$ **do**
  compute ϕ_a
  add $(a > \phi_a)$ to D'
end for

In words, for each default rule $(a > b)$ of D, we first compute the corresponding prime implicate ϕ_a. Then, $(a > \phi_a)$ is added to D'.

Now let us consider the computation of ϕ_a.

Algorithm 3

$\phi_a \leftarrow a \wedge b$
for all $(c > d) \in D$ **do**
  if $\vdash (c \rightarrow a)$ **and** $\vdash (d \rightarrow \neg b)$ **then**
    $\phi_a \leftarrow \phi_a \wedge \neg c$
  end if
end for

Algorithm 3 can be paraphrased as follows: by virtue of the definition of the truth conditions for $(a > b)$, b must appear in ϕ_a. The presence of a in ϕ_a stems from $facticity(a > a)$ and the definition of the truth conditions for $a > a$. Therefore, ϕ_a is initialized with $a \wedge b$. Then we add to ϕ_a the negation of each default rule $c > d$ of D such that the formulas $c \rightarrow a$ and $d \rightarrow \neg b$ are verified. This comes from *specificity*[2] and the definition of the truth conditions of $a > \neg c$. Finally, we get:

D'

$$\boxed{\begin{array}{l} bird(x) > fly(x) \wedge legs(x) \wedge bird(x) \wedge \neg penguin(x) \\ penguin(x) > \neg fly(x) \wedge penguin(x) \end{array}}$$

Soundness

Let us prove that the prime implicates computation is sound, namely:

$$(D' \models \phi > \psi) \rightarrow (D \models \phi > \psi) \tag{2}$$

with $\psi = \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_i \wedge \ldots \wedge \psi_n$ The proof is accomplished by induction on i.
Proof: Suppose that $(D' \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_{i-1}) \rightarrow (D \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_{i-1})$ Let us show that $D' \models \phi > \psi_i) \rightarrow (D \models \phi > \psi_i)$.
If $D' \models \phi > \psi_i$ then either (i) D contains the default rule $\phi > \psi_i$, hence $D \models \phi > \psi_i$; or (ii) D does not contain this default rule. In the latter case, either $\psi_i = \phi$ and *facticity* implies that $D \models \phi > \psi_i$ or $\psi_i = \neg\delta$ and $D \models \delta \rightarrow \phi$ and $D \models \delta > \chi$ and $D \models \phi > \neg\chi$, in which case *specificity* implies that $D \models \phi > \psi_i$

The truth conditions of defeasible rules $M, w \models \phi > \psi$ if and only if $*(w, [\phi]) \subseteq [\psi]$ imply that CE verifies the following rules:

CC: $(M, w \models \phi > \psi) \wedge (M, w \models \phi > \delta) \rightarrow (M, w \models \phi > \psi \wedge \delta)$

CM: $(M, w \models \phi > \psi \wedge \delta) \rightarrow (M, w \models \phi > \psi) \wedge (M, w \models \phi > \delta)$

CM implies that

$$\begin{array}{l}(D' \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_i) \rightarrow \\ \quad (D' \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_{i-1}) \wedge (D' \models \phi > \psi_i\end{array}$$

Thus

$$\begin{array}{ll}(D' \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_i) \rightarrow & \\ \quad (D \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_{i-1}) & \text{(by the inductive hypothesis)} \\ \quad \wedge (D \models \phi > \psi_i) & \text{(proved)}\end{array}$$

[2] Recall that the axiom which corresponds to *specificity* is $((p \rightarrow b) \wedge (b > f) \wedge (p > \neg f)) \rightarrow (b > \neg p)$

CC entails that

$$(D \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_{i-1}) \wedge (D \models \phi > \psi_i) \rightarrow \\ (D \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_i)$$

So $(D' \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_i) \rightarrow (D \models \phi > \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_i)$ □

4.2 Extensions computation

In the second part of the theorem prover, we have to compute the extensions of the theory which are maximal consistent sets. Each extension corresponds to a given ordering of the default rules. Thus if $ANT(D)$ contains n elements, $n!$ extensions have to be computed, which is the number of permutations of the elements.

Each extension E_i $(i \in [1..n!])$ is computed in the following way:

ALGORITHM 4
$E_i \leftarrow C$
for all $(a > \phi_a) \in D'$ **do**
  if ϕ_a is consistent with E_i **then**
    add $(a \rightarrow \phi_a)$ to E_i
  end if
end for

To illustate this algorithm, let us consider a more interesting example:

EXAMPLE 5

D

$bird(x) > fly(x)$
$bird(x) > legs(x)$
$penguin(x) > \neg fly(x)$

C

$penguin(x) \rightarrow bird(x)$
$penguin(tweety)$

The prime implicates computation leads us to build exactly the same set D' as in Example 4. Before computing the extensions, the default rules of D' have to be instantiated:

D'

$bird(tweety) > fly(tweety) \wedge legs(tweety) \wedge bird(tweety) \wedge \neg penguin(tweety))$
$penguin(tweety) > \neg fly(tweety) \wedge penguin(tweety)$

Next, 2! extensions have to be computed (as D' contains 2 default rules). For the first extension, E_1, the ordering is: $\zeta(bird) = 1$ and $\zeta(penguin) = 2$. First of all, E_1 is initiated with C.

$$fly(tweety) \wedge legs(tweety) \wedge bird(tweety) \wedge \neg penguin(tweety)$$

is not consistent with E_1, but $\neg fly(tweety) \wedge penguin(tweety)$ is, so

$$penguin(tweety) \rightarrow \neg fly(tweety) \wedge penguin(tweety)$$

is added to E_1:

E_1

$$\boxed{\begin{array}{l} penguin(x) \rightarrow bird(x) \\ penguin(tweety) \\ penguin(tweety) \rightarrow \neg fly(tweety) \wedge penguin(tweety) \end{array}}$$

For the next extension, E_2, the ordering is: $\zeta(bird) = 2$ and $\zeta(penguin) = 1$.

$$\neg fly(tweety) \wedge penguin(tweety)$$

is consistent with E_2, so

$$penguin(tweety) \rightarrow \neg fly(tweety) \wedge penguin(tweety)$$

is added to E_2, but

$$fly(tweety) \wedge legs(tweety) \wedge bird(tweety) \wedge \neg penguin(tweety)$$

is still not consistent with E_2, thus $E_2 = E_1$.

Non-monotonic consequence is defined in the following way:

DEFINITION 5 (Non-monotonic consequence)
$\Gamma \mid\!\sim \psi$ *iff for all* Γ*-extensions* $E, E \vdash \psi$

In Example 5, $\neg fly(tweety)$ is nonmonotonically inferred as it can be monotonically inferred from E_1 and E_2. This example illustrates the fact that the theorem prover supports the *Penguin Principle*.

EXAMPLE 6

D

$$\boxed{\begin{array}{l} quaker(x) > pacifist(x) \\ republican(x) > \neg pacifist(x) \end{array}}$$

C

$$\boxed{\begin{array}{l} quaker(Nixon) \\ republican(Nixon) \end{array}}$$

The prime implicates computation brings us to build the following D':

D'

$$\boxed{\begin{array}{l} quaker(x) > pacifist(x) \wedge quaker(x) \\ republican(x) > \neg pacifist(x) \wedge republican(x) \end{array}}$$

D' is now instantiated:

D'

$$\boxed{\begin{array}{l} quaker(Nixon) > pacifist(Nixon) \wedge quaker(Nixon) \\ republican(Nixon) > \neg pacifist(Nixon) \wedge republican(Nixon) \end{array}}$$

For the first extension, E_1, the ordering is:
$\zeta(quaker) = 1$ and $\zeta(republican) = 2$.

$$quaker(Nixon) \wedge pacifist(Nixon)$$

is consistent with E_1, so

$$quaker(Nixon) \rightarrow quaker(Nixon) \wedge pacifist(Nixon)$$

is added to E_1 but $republican(Nixon) \wedge \neg pacifist(Nixon)$ is not consistent with E_1.

E_1

$quaker(Nixon)$
$republican(Nixon)$
$quaker(Nixon) \rightarrow quaker(Nixon) \wedge pacifist(Nixon)$

For the second extension,$_2$, the ordering is:
$\zeta(quaker) = 2$ and $\zeta(republican) = 1$.

$$republican(Nixon) \wedge \neg pacifist(Nixon)$$

is this time consistent with E_2 so

$$republican(Nixon) \rightarrow republican(Nixon) \wedge \neg pacifist(Nixon)$$

is added to E_2 but $quaker(Nixon) \wedge pacifist(Nixon)$ is not consistent anymore with E_2.

E_2

$quaker(Nixon)$
$republican(Nixon)$
$republican(Nixon) \rightarrow republican(Nixon) \wedge \neg pacifist(Nixon)$

$pacifist(Nixon)$ can be monotonically inferred from E_1 but not from E_2. $\neg pacifist(Nixon)$ can be monotonically inferred from E_2 but not from E_1. Accordingly neither $pacifist(Nixon)$ nor $\neg pacificist(Nixon)$ is nonmonotonically inferred. This example illustrates the fact that the theorem prover supports the *Nixon Diamond*.

Remark

If the theory contains no embedded defaults then whenever we have to prove formulas in the nonmonotononic component of CE, those formulas are classical ones because proofs are made only on the right part or on the left part of the default rule but never on the whole default rule. Therefore in the first version of the implementation of the nonmonotonic part of CE, which does not allow embedded default rules, a theorem prover for the first order logic is used instead of a theorem prover for the monotonic component of CE. But if the theory contains embedded default rules then we will have to prove formulas which contain default rules. In that case a theorem prover for the monotonic part of CE is necessary.

The algorithm presented above depends on consistency checks in first order logic but first order consistency checking is undecidable. It is well known, however, that the purely universal fragment of first order logic is equivalent to zero order logic (no quantifiers). As zero order logic is decidable, the algorithm works only for the limited fragment of CE which does not contain any existential quantifiers.

To summarize, the algorithm presented above works only for the limited fragment of CE which does not contain any embedded default rules and any existential quantifiers. Hopefully, all the default rules stated so far belong to that fragment. The algorithm allowed us for instance to test the validity of the default rules involved in the spatiotemporal structure computation. However, embedded default rules will be necessary to model discourse phenomena other than spatiotemporal structure. As far as the extension of SDRT to dialogue is concerned, existential quantifiers appear to be needed (especially to model the semantic effects of discourse relations on mental states of the participants of a dialogue).

Complexity

The extension calculation takes a long time but that is not really surprising since it is a commonplace problem. In Algorithm 4 presented above, $n * n!$ consistency tests have to be performed to compute the extensions. However, as far as the discourse relation computation is concerned, the algorithm can be optimized because the default rules involved in the discourse relation computation belong to a subset of the language of CE. The set of those default rules has indeed the following property: the antecedent set and the consequent set are distinct. Moreover, those default rules do not contain any existential quantifiers. We have thus built an algorithm which takes into account those properties. Hopefully, it will be much more efficient as it performs n consistency tests. However, unfortunately, the algorithm can only be used for discourse relation computation.

5 Conclusion

We have built a theorem prover for the nonmonotonic logic, CE, which runs on a UNIX machine in Sicstus Prolog. This allowed us to implement the axioms and rules of SDRT for a fragment of natural language, namely narratives describing trajectories. This has been possible not only because we had this theorem prover, but also because a lot of theoretical results on lexical semantics, grammatical semantics, discourse pragmatics, discourse semantics and world knowledge have been achieved in my research group (Asher et al., 1994). It was not possible for space reasons to present this in that paper but as implementation issues go, the theorem prover was the most difficult one.

Our aim is neither to obtain a fully automated discourse analysis system, nor to get an efficient tool for real applications, but rather, to test the validity of the axioms and rules of SDRT which model the theoretical results. The axioms interact in a very complex way because of the non-monotonicity. It is thus very difficult to check by hand if they are correct. To get a fully automated discourse analysis, it is necessary to add a parser and a DRS builder. However, in order to check if the axioms and rules of SDRT are correct, we neither need a parser nor a DRS builder. This would unnecessarily make the system even more complicated. Another aim is to prove that the implementation is feasible, and, finally, if we want to get a more efficient system, we have to make approximations, so another aim of that implementation is to get a sound system to which approximations can be compared in order to see what the method is attempting to approximate.

The first version of the implementation of CE, which does not allow embedded default rules, was sufficient to test the axioms and rules stated so far. However, embedded defaults will be undoubtedly indispensable to model other inference patterns underlying discourse interpretation. This is why we are currently engaged in an implementation of CE which does allow embedded defaults. This involves building a theorem prover for the monotonic part of CE.

References

Asher, N. 1993a. Extensions for Commonsense Entailment. In *Proceedings of the Workshop on Conditionals*, Chambery, France. IJCAI.

Asher, N. 1993b. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publisher.

Asher, N., M. Aurnague, M. Bras, P. Sablayrolles, and L. Vieu. 1994. Computing the Spatiotemporal Structure of Discourse. In *Proceedings of the International Workshop on Computational Semantics*, pages 11–20, Tilburg, NL.

Asher, N. and M. Morreau. 1991. Commonsense Entailment : A Modal Theory of Nonmonotonic Reasoning. In *Proceedings of IJCAI*, Sydney Australia.

Daver, N. 1994. Mise en oeuvre informatique d'une théorie de représentation du discours (la SDRT), basée sur une logique non monotone. Master's thesis, Université Paul Sabatier, Toulouse, France.

Groenendijk, J. and M. Stokhof. 1991. Dynamic Predicate Logic. *Linguistics and Philosophy*, 14:39–100.

Grosz, B. and C. Sidner. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204.

Hobbs, J.R. 1991. Description of the Tacitus system as used for muc-3. In *Proceedings of Third Message Understanding Conference*, pages 200–206, San Mateo, California, May.

Kamp, H. and U. Reyle. 1993. *From Discourse to Logic : Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Kluwer Academic Publisher.

Lascarides, A. and N. Asher. 1991. Discourse Relations and Defeasible Knowledge. In *Proceedings of Meeting of the Association for Computational Linguistics*, pages 55–62, Berkeley CA.

Mann, W. and S. Thompson. 1987. Rhetorical Structure Theory : A Theory of Text Organization. Technical Report ISI-RS-87-190, Information Sciences Institute, Marina Del Rey, Ca, June.

Polanyi, L. 1988. A Formal Model of the Structure of Discourse. *Journal of Pragmatics*, 12:601–638.